AD-A052 951 CALIFORNIA UNIV SANTA CRUZ INFORMATION SCIENCES F/G 12/1

EFFICIENT SOLUTION OF SPACE-FILLING PUZZLES.(U)

FEB 78 W M MCKEEMAN, M J FAY, T J PENNELLO N00014-76-C-0682

UNCLASSIFIED TR-78-2-001 NL

| OF |

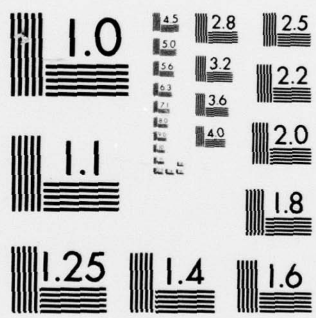
AD A052951

END

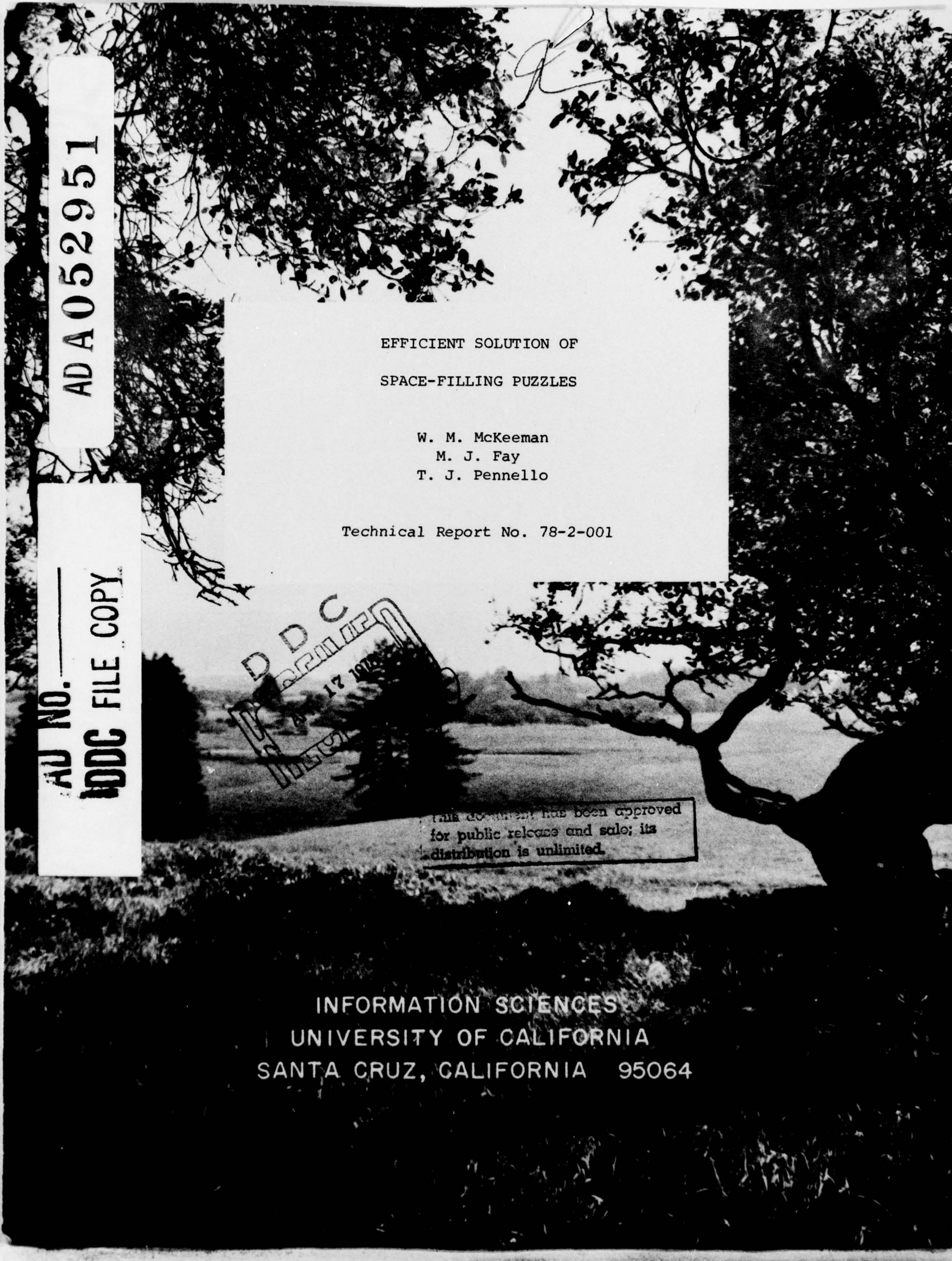
DATE FILMED

5-78

DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



EFFICIENT SOLUTION OF

SPACE-FILLING PUZZLES

W. M. McKeeman
M. J. Fay
T. J. Pennello

Technical Report No. 78-2-001

INFORMATION SCIENCES
UNIVERSITY OF CALIFORNIA
SANTA CRUZ, CALIFORNIA 95064

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 TR-78-2-001	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9
4. TITLE (and Subtitle) 6 EFFICIENT SOLUTION OF SPACE-FILLING PUZZLES.		5. TYPE OF REPORT & PERIOD COVERED Technical rept.,
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 William M. McKeeman, M. J. Fay T. J. Pennello		8. CONTRACT OR GRANT NUMBER(s) 15 ONR N00014-76-C-0682
9. PERFORMING ORGANIZATION NAME AND ADDRESS William M. McKeeman and Sharon Sickel Information Sciences, UCSC, Rm. 239 AS Santa Cruz, California 95064		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE 11 Feb[redacted]78
		13. NUMBER OF PAGES 9
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research University of California 553 Evans Hall Berkeley, California 94720 12 13p.		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Space-filling, group theory, puzzles, integer programming, backtrack, Soma Cube, Instant Insanity, exact cover.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Eight-queens, the Mutilated Checkerboard of Golomb, Instant Insanity, and the various figures that can be built out of the seven Soma pieces are all examples of space-filling puzzles. Such puzzles may be described in terms of sets and groups. Puzzle solution is then seen to be a special case of finding an exact cover for a set. The latter can be solved by backtrack algorithms. Certain well-known combinatorics-reducing counting arguments applicable to the puzzles are formalized relative to the

(OVER)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

410 350 LB

20. ABSTRACT (continued)

exact cover problem. Then the special group structure of the space-filling puzzles is used to choose specific counting arguments and to pick representatives for symmetrically equivalent solutions.

EFFICIENT SOLUTION OF

SPACE-FILLING PUZZLES

W. M. McKeeman

M. J. Fay

T. J. Pennello

Information Sciences

University of California at Santa Cruz

This work supported in part by the Office of Naval Research under Contract N00014-76-C-0682

EFFICIENT SOLUTION OF
SPACE-FILLING PUZZLES

W. M. McKeeman
M. J. Fay
T. J. Pennello

Information Sciences
University of California at Santa Cruz

Keywords and phrases: Space-filling, group theory, puzzles, integer programming, backtrack, Soma Cube, Instant Insanity, exact cover.

CR categories: 3.15, 3.64, 5.39, 5.41.

This work supported in part by the Office of Naval Research under contract 76-C-0682.

Abstract

The Eight-queens, the Mutilated Checkerboard of Golomb, Instant Insanity, and the various figures that can be built out of the seven Soma pieces are all examples of space-filling puzzles. Such puzzles may be described in terms of sets and groups. Puzzle solution is then seen to be a special case of finding an exact cover for a set. The latter can be solved by backtrack algorithms. Certain well-known combinatorics-reducing counting arguments applicable to the puzzles are formalized relative to the exact cover problem. Then the special group structure of the space-filling puzzles is used to choose specific counting arguments and to pick representatives for symmetrically equivalent solutions.

1. Introduction

The Soma Cube [3,5,6,9,10,14,16,17,25], Instant Insanity [4,17] and Eight Queens [12,17,26] are all examples of space-filling puzzles. There are others [8,11,13,15,19,20,22,23,24]. In general terms each puzzle consists of a set of pieces and the challenge is to build a specific figure or configuration out of them.

For example, the seven Soma Cube pieces are shown in Figure 1; some of the puzzles that one can attempt to construct are shown in Figure 2.

It is straightforward to formulate exhaustive algorithms for the puzzles. Our primary objective is to improve the efficiency of these algorithms. We will do this by a preliminary analysis that has the effect of pruning the search tree before the search begins. Such pruning can in general benefit both backtracking and heuristic search algorithms.

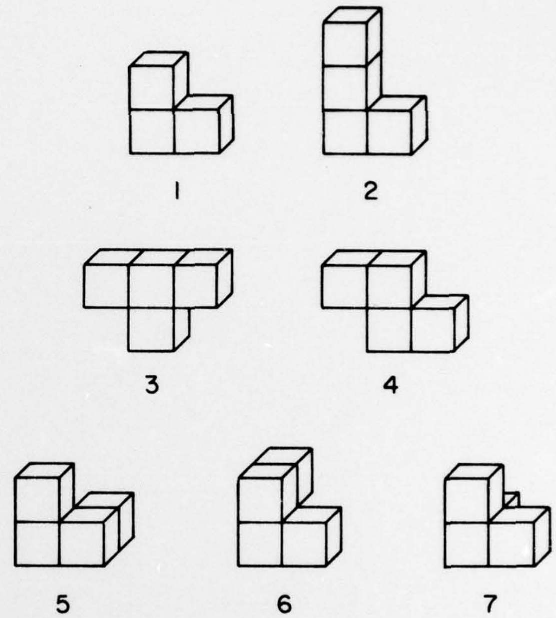


Figure 1: Pieces of the Soma Cube

The problem is interesting relative to Simon's definition of well-structuredness [19]. The basic components of this definition are (1) test for solution state, (2) characterization of start, goal and intermediate states, (3) representation of necessary state transitions, (4) representation of knowledge about the problem, (5) real-world accuracy of the model, (6) practical amounts of computation required.

The problem of space-filling puzzles can be formalized to the first five criteria. As we expose previously hidden structure in the problem space, we progress in measurable steps toward satisfaction of the sixth component. In the more general sense of Simon, well-structuredness is seen as a relation between the problem and solver. As the solver gains knowledge, the amount of apparent structure increases.

The steps to solution are:

(1) Describe the puzzles in terms of groups and sets.

(2) Formalize certain well-known counting arguments in terms of the exact cover problem [1].

(3) Use the groups to pick particular counting arguments specifically applicable to puzzles, and

(4) Pick a representative for each class of symmetrically equivalent solutions.

2. A Formal Solution

Space-filling puzzles can be described in terms of the component pieces, the figure to be constructed, and the ways pieces can be moved in space.

One has successfully constructed a figure when certain conditions have each been satisfied exactly once. The Soma Cube is the 3 x 3 x 3 cube in Figure 2. To build it, one must satisfy 34 conditions: each of the seven pieces must be used and each of the 27 unit cubes in the figure must be occupied. Thus the set of conditions can be used to describe the figure to be built.

A piece can be described by the set of conditions it satisfies. Such a description is not unique; the piece satisfies different conditions when it is in different positions in the figure.

The moves in space that carry pieces from one position to another may be thought of as functions over the set of conditions. Thus the givens are:

(1) Figure: a finite set of conditions to be satisfied for a particular puzzle.

(2) Pieces: a set of subsets of Figure, each subset describing a different piece, in a particular position.

(3) Moves: a set of functions carrying pieces from one position to another.

Underlying the figure there is sometimes a larger (perhaps even infinite) set of conditions describing a geometric space-filling. A space-filling is a regular pattern of objects that fit together to fill up the entire space in which they are embedded. For the Soma Cube it is the infinite set of unit cubes filling three-dimensional Cartesian space. Another example is given in Figure 3. There are many such examples, in higher dimensional spaces, and even non-Euclidean spaces.

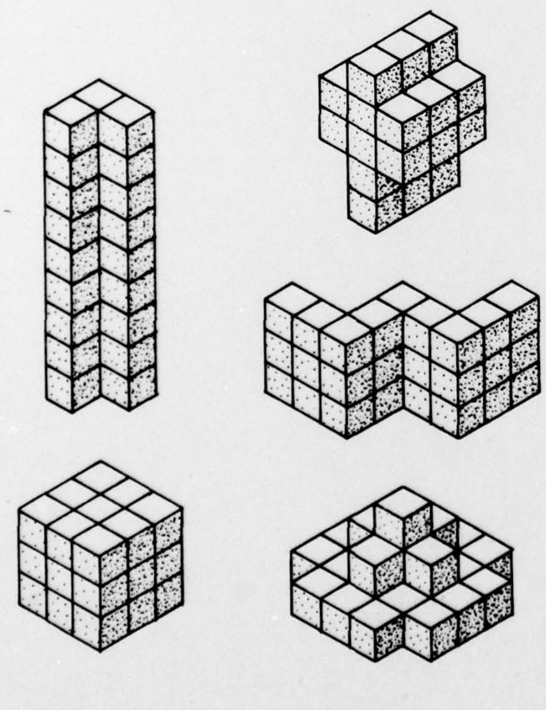


Figure 2. Puzzles for the Soma Cube Pieces

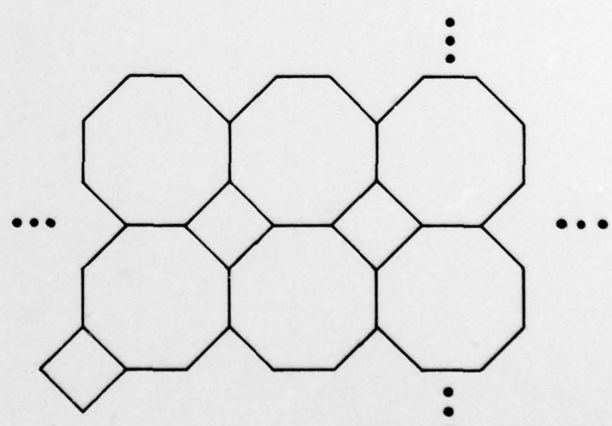


Figure 3: A Space-filling

The functions in Moves are permutations on this larger set and are closed under the operation of functional composition; thus they form a group. For the Soma Cube, Moves consists of translations, rotations and reflections in 3-space.

A puzzle may then be formally solved as follows:

(1) Use Moves to generate all possible placements of pieces.

(2) Find subsets of placements that
  (a) fill Figure and
  (b) don't intersect.

(3) Collect solutions that can be mapped onto each other by Moves.

That is, the solution is the set

$$\{\text{Moves}(S) \mid S \subseteq \cup\text{Moves}(\text{Pieces}), \cup S = \text{Figure}, \text{disjoint}(S)\}^{\S} \qquad (2.1)$$

Because Moves places pieces throughout the entire space-filling, some of the sets may be infinite. A finite version can be given in terms of the symmetry subgroup of Moves. It is given by the set of functions:

$$\text{Symmetry} = \{f \mid f \in \text{Moves}, f(\text{Figure}) = \text{Figure}\}.$$

---

§ Let S be a set of objects in the domain of a function f. Then it is convenient to extend f to apply to S, yielding a set of values: $f(S) \overset{\text{def}}{=} \{f(s) \mid s \in S\}$. Note that this definition is applicable to sets of sets, and so on. Further, let F be a set of functions, then $F(x) \overset{\text{def}}{=} \{f(x) \mid f \in F\}$.

E.g., $\text{Moves}(\text{Pieces}) = \{\{m(p) \mid p \in \text{Pieces}\} \mid m \in \text{Moves}\}$.

For any set of sets S, $\text{disjoint}(S) \overset{\text{def}}{=} (\forall s,t \in S)\ s \neq t \text{ implies } s \cap t = \phi$; and $\cup S \overset{\text{def}}{=} \bigcup_{s \in S} s$. The effect of $\cup$ is to remove one level of set bracketing, e.g., $\cup\{\{1,2\}, \{4,\{5\}\}\} = \{1,2,4,\{5\}\}$.

Then we get a new solution formula:

$$\{\text{Symmetry}(S) \mid S \subseteq (\cup\text{Moves}(\text{Pieces})) \cap 2^{\text{Figure}}, \cup S = \text{Figure}, \text{disjoint}(S)\}^{\ddagger} \qquad (2.2)$$

It is formula 2.2 that yields an exhaustive backtracking algorithm and serves as a basis for the improved versions to be developed. The natural one-to-one correspondence between 2.1 and 2.2 is given in [18].

3. Applying the Formal Model

The elements in Figure may be given individual names, and the rest of the givens defined in terms of them. The functions in Moves need be only partially specified since a move outside of Figure can be ignored. Furthermore we need only provide Generators for the group of Moves since they can be iterated over Pieces to get all that lie within Figure. Popular puzzles fall into two classes: Those where one kind of piece is used as often as necessary, and those where there is a fixed set of individual pieces each to be used exactly once. Both fit within the formalism but are represented somewhat differently.

Puzzle 1 (Mutilated Checkerboard)
How many different ways can a 4 x 4 checkerboard missing two diagonally opposite corners be covered with dominoes the size of two squares [11]?

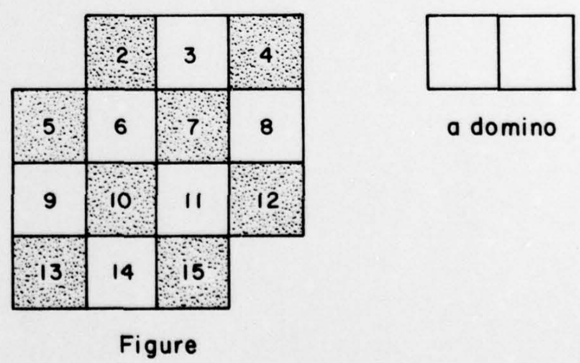


Figure.4: Mutilated Checkerboard

---

‡ $2^{\text{Figure}} = \{p \mid p \subseteq \text{Figure}\}$.

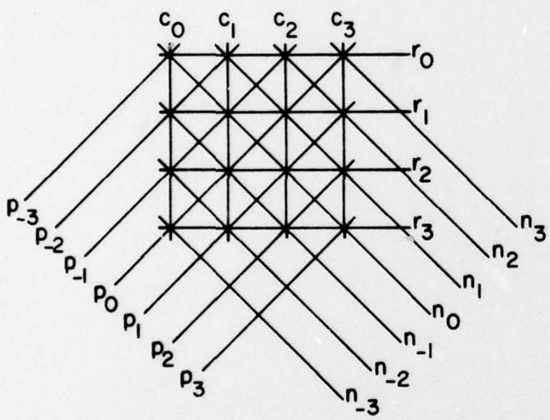
Figure = $\{2,3,\ldots 14,15\}$

Pieces = $\{\{2,3\}\}$.

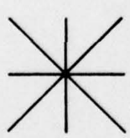
Generators = {translate,rotate} where

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
translate(x)†	2	3	4	Ω	6	7	8	Ω	10	11	12	Ω	14	15	16	Ω
rotate (x)	4	8	12	16	3	7	11	15	2	6	10	14	1	5	9	13

<u>Puzzle 2</u> (N queens) How many different ways can N queens be placed on an N x N chessboard so that no pair is mutually threatening? The representation is in terms of the files of queen attack (c = column, r = row, p = positively sloping diagonal, n = negatively sloping diagonal). Once N queens are placed, some diagonals are still unthreatened,



a 4 X 4 chessboard



a queen's space

Figure 5: N Queens

† The condition Ω stands for "outside the puzzle."

thus we add a "free" diagonal to Pieces.

Figure = $\{c_i \mid 0 \le i < N\} \cup \{r_i \mid 0 \le i < N\}$

$\cup \{p_i \mid -N < i < N\} \cup \{n_i \mid -N < i < N\}$

Pieces = $\{\{c_0, r_0, p_{-N}, n_0\}, \{p_0\}\}$

Generators

x		c_i	p_i	n_i	r_i
translate(x)	for i<N	c_{i+1}	p_{i+1}	n_{i+1}	r_i
	for i=N	Ω	Ω	Ω	r_i
rotate(x)		r_i	n_{-i}	p_i	c_{N-i}
reflect(x)		c_{N-i}	n_{-i}	p_{-i}	r_i

Notice we included reflections this time, giving a larger symmetry group and leading to fewer non-equivalent solutions.

How many different ways N^M hyper-rooks can be placed on an M+1 dimensional hyper-chessboard of side N is similar to the N queens problem. There are several formalizations of the problem, each with a different view of symmetry. Some have been solved in closed form [15]. Others are known to be very difficult (for example, the N^2 hyper-rooks problem is equivalent to enumerating the N x N Latin squares [2]). These problems can also be expressed as space-filling puzzles.

<u>Puzzle 3</u> (The Soma Cube) How many different ways can a 3 x 3 x 3 cube (or any other object in Figure 2) be built out of the seven pieces in Figure 1?

Place a corner of the puzzle at the origin in 3-space, and let the coordinate triples of each cube describe it. The pieces themselves are numbered 1 - 7.

Figure =

{000,001,002,010,011,012,020,021,022,
100,101,102,110,111,112,120,121,122,
200,201,202,210,211,212,220,221,222,
1, 2, 3, 4, 5, 6, 7}

Pieces =

{{000,100,010,1},
{000,100,200,010,2},
{000,100,200,110,3},
{000,100,110,210,4},
{000,100,010,101,5},
{000,100,010,001,7}}.

Generators contains two rotations, a translation, and a reflection similar to that of Puzzle 1. The piece names 1 - 7 are invariant under the moves except that the reflection of 5 is 6 and vice versa.

Puzzle 4 (Instant Insanity) The puzzle consists of four colored cubes. The solution is achieved when the cubes are aligned (say stacked up) so that each of the four colors appears on each side of the stack. The sides that happen to be faced up or down are ignored. The basic objects out of which one builds Instant Insanity are color/direction pairs and piece names. Let the colors be {r,w,b,g} (meaning red, white, blue, green), and the four directions be {N,E,S,W}, and the piece names be {1,2,3,4}. Then

Figure = {rN,rE,rS,rW,wN,wE,wS,wW,
bN,bE,bS,bW,gN,gE,gS,gW,
1,2,3,4}

Each original colored cube has three initial positions in Pieces corresponding to each opposite pair of faces in the ignored (vertical) orientation. (Unfortunately the published literature on Instant Insanity and some commercial versions of the puzzle do not have the same coloring. The one given here is taken from Brown's paper [4].)

Pieces =

{{rN,bE,rS,gW,1},{bN,bE,wS,gW,1},{rN,wE,rS,bW,1},
{gN,wE,bS,rW,2},{gN,wE,wS,rW,2},{gN,wE,bS,gW,2},
{gN,wE,rS,bW,3},{rN,wE,wS,bW,3},{gN,wE,rS,rW,3},
{gN,wE,gS,bW,4},{rN,wE,gS,bW,4},{gN,gE,gS,rW,4}}.

Generators is given by a rotation mapping N to E to S to N, and a reflection (corresponding to a rotation exchanging vertical faces, in the actual puzzle) mapping N to S to N.

4. Counting Arguments

Given an arbitrary collection of sets B, the set of exact covers is given by

$$\{S \mid S \subseteq B,\ \cup S = \cup B,\ \text{disjoint}(S)\} \quad (4.1)$$

Since $\cup(\cup\text{Moves}(\text{Pieces}) \cap 2^{\text{Figure}}) = \text{Figure}$[§] in formula 2.2, it is a special case of 4.1 (except for the further collection of solutions into equivalence classes). The exact cover problem is NP-complete [1]. It can be solved in exponential time by backtracking. We introduce it here to formalize some counting arguments that can potentially reduce, or even eliminate the backtracking search for puzzle solutions.

We first introduce the concepts of a solution vector for an exact cover S, and a characteristic vector for an element of B. Both kinds of vectors are dependent on a partitioning of $\cup B$. We can compute the characteristic vectors of B directly, use them to compute the solution vectors of the exact covers S, and then use the results to reformulate the search algorithm.

Let $\{t_1,t_2,\ldots,t_n\}$ be an arbitrary disjoint partitioning of $\cup B$. The *characteristic vector, c, of an element, b, of* B relative to the partitioning $\{t_1,\ldots,t_n\}$ is defined by

$$c = \| \, |b \cap t_1|, |b \cap t_2|, \ldots |b \cap t_n| \, \|.$$

Now partition B itself according to the characteristic vectors (that is, $B = B_1 \cup B_2 \ldots \cup B_m$ where all b in the same B_i have the same characteristic vector).

We define the solution vector, x, of an exact cover S by

$$x = \| \, |S \cap B_1|,\ |S \cap B_2|, \ldots |S \cap B_m| \, \|.$$

The trick is to compute the set X of solution vectors x before the solutions S are known.

If we construct the matrix $\|c_{ij}\|$ where the j^{th} column is the characteristic vector c corresponding to B_j, then the solutions x in non-negative integers of the equations

$$\sum_{j=1}^{m} c_{ij} x_j = |t_i| \quad (4.2)$$

are the needed solution vectors (see [18] for proof). If X is the set of solutions to 4.2, then formula 4.3 gives the same result as formula 4.1.

$$\bigcup_{x \in X} \{S \mid S = \bigcup_{j=1}^{m} S_j \text{ where } S_j \subseteq B_j, \quad (4.3)$$
$$|S_j| = x_j,\ \cup S = \cup B,\ \text{disjoint}(S)\}$$

The effect is to break the main problem into smaller ones, each with smaller spaces to search. This results, in some cases, in a substantial reduction of computing time.

§ A poorly-formed puzzle definition might not have moves or pieces sufficient to cover Figure, in which case there are trivially no solutions.

Suppose the smaller problems of formula 4.3 are still too difficult. Pick a second partitioning $\{t_1',\ldots,t_{n'}'\}$, getting a second partitioning $\{B_1',\ldots,B_{m'}'\}$ of B and a second set of vectors X'. Then formula 4.3 generalizes to

$$\bigcup_{\substack{x\in X \\ x'\in X'}} \{S \mid S = \bigcup_{\substack{1\le i\le m \\ 1\le j\le m'}} S_{ij} \text{ where } S_{ij} \subseteq B_i \cap B_j',\tag{4.4}$$

$$\sum_{i=1}^{m} |S_{ij}| = x_j' \;,\; \sum_{j=1}^{m'} |S_{ij}| = x_i \;,$$

$$\cup S = \cup B,\ \text{disjoint}(S)\} \;.$$

Both the sets $B_i \cap B_j'$ and S_{ij} are smaller, continuing to restrict each of the subproblem spaces. This process can be continued profitably until the increase in the number of subproblems dominates the computation.

5. Applying Counting Arguments

There is no reason to believe that it is possible in general to pick partitionings of $\cup B$ that lead to an efficient search. Nevertheless there are good partitionings for popular puzzles [3,4,11, 25]. This seems to be due to the structure imposed on B by the group of moves that generates it:

$$B = (\cup \text{Moves(Pieces)}) \cap 2^{\text{Figure}} \;.$$

In terms of the givens of the space-fillings, formula 4.3 becomes

$$\bigcup_{x\in X} \{\text{Symmetry}(S) \mid S = \bigcup_{j=1}^{m} S_j \text{ where}$$

$$S_j \subseteq B_j,\ |S_j| = x_j \;,\tag{5.1}$$

$$\cup S = \text{Figure},\ \text{disjoint}(S)\}$$

Let M be an arbitrary subgroup $M \subseteq$ Moves. Then the set

$$\{M(a) \cap \text{Figure} \mid a \in \text{Figure}\}\tag{5.2}$$

is a disjoint partitioning of Figure and can be used in the counting arguments.

Suppose, for example, we take the diagonal translation subgroup of Moves for puzzle 1. Then we get

$\{t_1,t_2,t_3,t_4\} =$

$\{\{1,3,6,8,9,11,14,16\},$

$\{2,4,5,7,10,12,13,15\}\}$

and characteristic vector

$$\begin{Vmatrix} 1 \\ 1 \end{Vmatrix}$$

and attempt to solve

$$\begin{Vmatrix} 1 \\ 1 \end{Vmatrix} \cdot x = \begin{Vmatrix} 6 \\ 8 \end{Vmatrix}$$

in non-negative integers. There are no such 1 x 1 vectors x. Thus formula 5.1 tells us there are <u>no</u> solutions to the puzzle.

Some "common sense" arguments used in solving puzzles drop out of formula 5.1. Suppose M = Moves for Puzzle 3, the Soma Cube. Then the partition of Figure is

$\{\{1\},\{2\},\{3\},\{4\},\{5,6\},\{7\},\{000,\ldots,222\}\}$.

The characteristic vectors give, for equation 4.2,

$$\begin{Vmatrix} 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \\ 3&4&4&4&4&4 \end{Vmatrix} \cdot x = \begin{Vmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 1 \\ 27 \end{Vmatrix}$$

which has the single solution $x = \begin{Vmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 1 \end{Vmatrix}$.

Upon examining the corresponding sets B_i, one finds that the algorithm will attempt to use only one each of pieces 1,2,3,4,7 and two from a set where pieces 5 and 6 are combined.

Similarly, let M = Moves for Puzzle 1. We get the trivial partitioning of Figure

$\{\{2,3,4,5,6,7,8,9,10,11,12,13,14,15\}\}$

which leads to the single equation 4.2

$2x = 14$, or $x = 7$

telling us to use exactly seven pieces to attempt to solve the puzzle.

While we do not know how to derive a good partitioning of Figure, we do know when a partitioning is good. The criterion is that the number of trial solutions in formula 5.1 is small. Specifically, not too many vectors x, small (preferably zero) components x_j, and where they are non-zero, small corresponding sets B_j.

The partitions generated by subgroups (as in 5.2) distribute the piece placements fairly uniformly among the B_j, thus preventing any one of them from being very large. In fact a related partitioning $\{B_1{}'\ldots B_{n'}{}'\}$ of B can be computed directly from the subgroup. Let

$$\{B_1',\ldots B_{n'}'\} = \{M(p) \cap 2^{\text{Figure}} \mid p \in \cup\text{Moves(Pieces)}\}.$$

Then each $B_k' \subseteq B_j$ for some j. That is, the characteristic vector is constant over the elements of each $B_k{}'$ (proof in [18]). We can then compute $\|c_{ik}{}'\|$ from the partitioning (instead of vice versa) and carry out the computations of formulas 4.2 and 5.1.

For example, let M be the rotation subgroup (order 24) of Moves for the Soma Cube about its center. The partitioning $\{t_1,\ldots,t_n\}$ of Figure is

{{1},{2},{3},{4},{5},{6},{7},	piece names
{000,002,020,022,200,202, 220,222},	vertices
{100,010,001,102,012,021, 120,210,201,122,212,221}	edges
{110,101,011,112,121,211}	faces
{111} }	center

and the equation 4.2 is

$$\left\|\begin{array}{cccccccccccccccccc}
1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&1&1&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&0&0&1&1&2&2&0&0&0&2&0\\
0&1&1&2&1&1&2&1&2&2&2&2&2&0&2&3&1&1\\
2&1&2&1&1&1&0&2&2&1&1&0&0&3&1&1&1&2\\
1&1&0&0&0&0&0&1&0&0&0&0&0&1&1&0&0&1
\end{array}\right.$$

$$\left.\begin{array}{cccccccccccccc}
0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
1&1&0&0&1&1&0&0&1&1&0&0&1&1\\
2&2&1&2&1&2&1&2&1&2&0&2&1&3\\
1&1&2&2&1&1&2&2&1&1&3&1&2&0\\
0&0&1&0&1&0&1&0&1&0&1&1&0&0
\end{array}\right\| \cdot x = \left\|\begin{array}{c}1\\1\\1\\1\\1\\1\\1\\8\\12\\6\\1\end{array}\right\|$$

The principal difference between this solution and the one getting $\|c_{ij}\|$ directly is that four columns of $\|c_{ij}\|$ are duplicates. That does not, however, have any substantial effect on the time to solution so long as the equation solver does not duplicate effort in dealing with them.

There are 72 solution vectors x. If there were no column duplications there would be 11 vectors. In both cases the coefficients x_j are zero or one. The computation of formula 5.1 takes the same amount of effort in either case.

As an indication of the effectiveness of this counting technique, we can compare the total number of possible configurations allowed, with and without our counting analysis. For the Soma Cube, using each piece once, we have $144 \times 144 \times 72 \times 72 \times 96 \times 96 \times 64 = 6.3 \times 10^{13}$ possible assemblies. But using our analysis as in the previous paragraphs, the theoretical maximum becomes the sum of 11 products with smaller factors, yielding a total of 1.2×10^{11} assemblies. This gain of two orders of magnitude can be further enhanced by elimination of symmetry and by efficient backtracking schemes.

6. Elimination of Symmetry

In formula 2.2 we have for each class computed, $|\text{Symmetry(S)}| \geq 1$. It is sufficient to compute a single representative of each class where it is greater than one. We can, at the same time, reduce the search space and therefore computation effort. The approach is to carry out the counting argument with the subgroup M of section 5 equal to the symmetry group, and then use the information to change each subproblem in formula 5.1 into a simpler, less symmetric, space-filling puzzle. We can then iterate the process if necessary.

Let M = Symmetry. Compute the corresponding partition of Figure from formula 5.2, and compute the vectors x from formula 4.2. Each vector is treated separately. Pick one. Suppose there is a k such that $x_k = 1$ and $|B_k| = |\text{Symmetry}|$. Pick a $b \in B_k$. Then if we discard B_k, the coefficient x_k, and compute

$$\{S' \mid S' = \bigcup_{j\neq k} S_j,\ \text{where } S_j \subseteq B_j,\ |S_j| = x_j,$$
$$\cup S' = \text{Figure-b} \quad \text{disjoint}(S')\},$$

it is as though we had started out to solve for Figure-b without piece placements B_k. The sets $S' \cup \{b\}$ are the

needed solution representatives, for a given solution vector x. None of these representatives are equivalent to solution representatives for another x.

Suppose $x_k = 1$ but $|B_k| < |\text{Symmetry}|$. We can carry out the above but stop short of computing the S' since the $S' \cup \{b\}$ will not necessarily be unique representatives. Instead we carry out the counting argument again with subgroup M given by the intersection of Symmetry with the symmetry group of Figure - b. We continue in this manner until the symmetry is removed or we exhaust Figure.

Suppose, finally, that there is no $x_k = 1$. Pick any $x_k > 1$. We shall now have to treat subsets of B_k of size x_k as piece placements to carry out the symmetry reduction above. Compute

$$\{\text{Symmetry}(H) \mid H \subseteq B_k, |H| = x_k, \text{disjoint}(H)\}$$

and then for each member we replace B_k with a set of "super pieces"

$$B_k \leftarrow \{\cup G \mid G \in \text{Symmetry}(H)\}$$

and set

$$x_k \leftarrow 1.$$

It is as though we had to set out to solve the original puzzle but with the contributions from B_k already stuck together. We can then apply the preceding algorithms to remove the symmetry.

For the Soma Cube, the symmetry group has order 48 and so do the sets B_k for piece 2. Thus symmetry can be eliminated in a single step by solving for the puzzle less piece 2, at the same time reducing the computational effort by a factor of 48.

7. Conclusions

We have developed a model of space-filling puzzles and given algorithms for their solutions. The central problem is partitioning the puzzle to yield useful counting arguments. It was shown that some guidance for the choice can be taken by considering subgroups of the set of spatial moves. The same mechanism can be used for an efficient removal of unwanted symmetry. It is this dependence on groups of spatial moves that makes space-filling puzzles more easily solvable than general exact set cover problems.

There remain some interesting questions. Can the problem be better solved directly in terms of the (perhaps infinite) model of equation 2.1?

What is the optimal sequence of partitionings leading to a minimal overall effort [7] in using formula 4.4? And, what is its relation to the subgroups M of section 5?

The matrices $\|c_{ij}\|$ appear to have very special and easily solved form. Can this be formalized to yield a more efficient equation solver for this class of problems?

The group Moves arises from geometrical models. Can one characterize them in a way that simplifies the task?

Acknowledgements

A large number of people have made suggestions, and worked with the authors on this problem. Cleve Moler suggested formulating the algorithm in terms of sets in 1962. Todd Hirozawa implemented the first program at Santa Cruz to utilize the finite model and some of the counting arguments mentioned above. Conversations with Ken Friedenbach led to the formalization in terms of groups. Donald Knuth, Tom Atwater, Jim Horning, Sharon Sickel, Donald Michie, Martin Gardner and an unnamed referee were very helpful in commenting on earlier drafts of this material and pointing out important references. Many others, most especially the graduate students at Santa Cruz, have contributed in ways that were less specific by no less important. We thank them all.

References

1. Aho, A. V., J. E. Hopcroft and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley (1974).

2. Alter, R., "How Many Latin Squares Are There?", American Math. Monthly, Vol. 82, No. 6 (June-July 1975), p. 632.

3. Atwater, T. (Ed.), Soma Addict, Vol 1, No. 1 & 2, Vol 2, No. 1 & 2, Parker Bros., Inc., Salem, MA (1970)

4. Brown, T. A., "A Note on 'Instant Insanity'", Mathematics Magazine (Sept.-Oct. 1968), p. 167.

5. Chichelli, R. J., S. L. Gulden, "Another Solution to the Soma Cube Puzzle," SIGPLAN Notices (Oct., 1975), p. 11.

6. DeLong, R., "Solutions to the Soma Cube Problem," SIGPLAN Notices (Oct., 1974), p. 7.

7. Fillmore, J. P. and S. G. Williamson, "On Backtracking: A Combinatorial Description of the Algorithm", SIAM J. Comput., Vol. 3, No. 1 (Mar. 1974), p. 41.

8. Fletcher, J., "A Program to Solve the Pentomino Problem by Recursive Use of Macros", Comm. ACM, Vol. 8, No. 10 (Oct. 1965), p. 621.

9. Gardner, M., "Mathematical Games", Scientific American (Sept. 1972, Oct. 1972, Oct. 1973).

10. _____, "The Soma Cube", in Mathematical Puzzles and Recreations, Chapter 6, Simon and Schuster (1961).

11. Golomb, S., Polyominoes, Scribners (1965).

12. Hollander, D. H., "An Unexpected Two-Dimensional Space-group Containing Seven of the Twelve Basic Solutions to the Eight-queens Problem", J. Rec. Math., Vol. 6, No. 4 (Fall 1973), p. 287.

13. Kawai, S., K. Noshita, I. Takeuchi, "On Backtrack Programming and Some Results on Combinatoric Puzzles", Proc. Second USA-Japan Computer Conference (Aug. 1975), p. 300.

14. Knuth, D. E., estimations of the run time for backtracking solutions for the Soma Cube, The Art of Computer Programming. Vol. 4, Addison-Wesley (to appear).

15. Larson, L. C., "Essentially Different Nonattacking Rook Arrangements," J. Rec. Math., Vol. 7, No. 3 (Summer 1974), p. 178.

16. McKeeman, W. M., "Solving Space-filling Puzzles", CEP Reports, Vol. 6, No. 1., The University of California at Santa Cruz, CA 95064 (May 1974).

17. McKeeman, W. M., "A Formal Model for Space-filling Puzzles", Machine Intelligence 8 (D. Michie, T. Elcock, Eds.)(1977).

18. McKeeman, W. M., "Proofs on Space-filling Puzzles," unpublished notes, U.C. Santa Cruz (1977).

19. Meeus, J., "Some Polyomino and Polyamond Problems", J. Rec. Math, Vol. 6, No. 3 (Summer 1973), p. 215.

20. _____, "Tetracubes," J. Rec. Math., Vol. 6, No. 4 (Fall 1973), p. 257.

21. Simon, H. A., The Structure of Ill Structured Problems, Artificial Intelligence 4 (1973) p. 181.

22. Torbijn, I. P. J., "The Unknown World of Octaimonds", J. Rec. Math. Vol. 7, No. 1 (Winter 1974), p. 1.

23. Wagner, N. R., "Construction with Pentacubes", J. Rec. Math. Vol. 5, No. 4 (Fall 1972), p. 266.

24. _____, "Construction with Pentacubes-2", J. Rec. Math. Vol. 6, No. 3 (Summer 1973), p. 211.

25. Whinihan, M., and C. Trigg, "Parity and Centerness Applied to the Soma Cube", J. Rec. Math., Vol. 6, No. 1 (Winter 1973), p. 61.

26. Wirth, N. E., "Program Development by Stepwise Refinement," Comm. ACM Vol. 14, No. 4 (Apr. 1971), p. 221.

OFFICIAL DISTRIBUTION LIST

Contract N00014-76-C-0682

Defense Documentation Center
Cameron Station
Alexandria, VA 22314
12 copies

Office of Naval Research
Information Systems Program
Code 437
Arlington, VA 22217
2 copies

Office of Naval Research
Code 102IP
Arlington, VA 22217
6 copies

Office of Naval Research
Code 200
Arlington, VA 22217
1 copy

Office of Naval Research
Code 455
Arlington, VA 22217
1 copy

Office of Naval Research
Code 458
Arlington, VA 22217
1 copy

Office of Naval Research
Branch Office, Boston
495 Summer Street
Boston, MA 02210
1 copy

Office of Naval Research
Branch Office, Chicago
536 South Clark Street
Chicago, IL 60605
1 copy

Office of Naval Research
Branch Office, Pasadena
1030 East Green Street
Pasadena, CA 91106
1 copy

New York Area Office
715 Broadway - 5th Floor
New York, NY 10003
1 copy

Naval Research Laboratory
Technical Information Division
Code 2627
Washington, DC 20375
6 copies

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps (CodeRD
Washington, D. C. 20380
1 copy

Naval Electronics Laboratory Center
Advanced Software Technology Division
Code 5200
San Diego, CA 92152
1 copy

Mr. E. H. Gleissner
Naval Ship Research & Development Cent
Computation and Mathematics Department
Bethesda, MD 20084
1 copy

Captain Grace M. Hopper
NAICOM/MIS Planning Branch (OP-916D)
Office of Chief of Naval Operations
Washington, D. C. 20350
1 copy

Mr. Kin B. Thompson
Technical Director
Information Systems Division (OP-911G)
Office of Chief of Naval Operations
Washington, D. C. 20350
1 copy